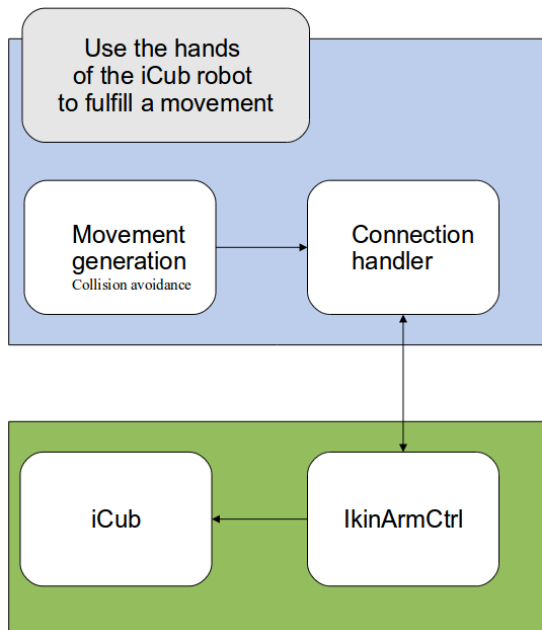


RobotDoc 2010

Katrin Lohan, Andre Lemme, Arne Nordmann

26.10.2010

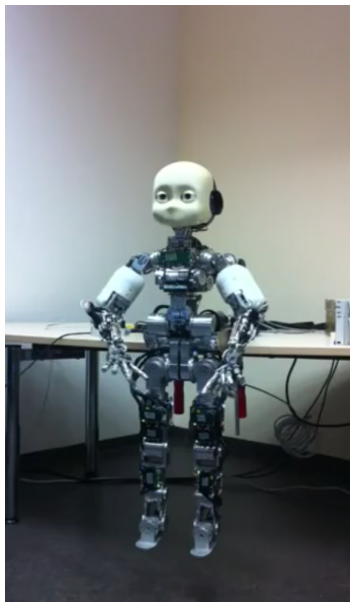
The Job



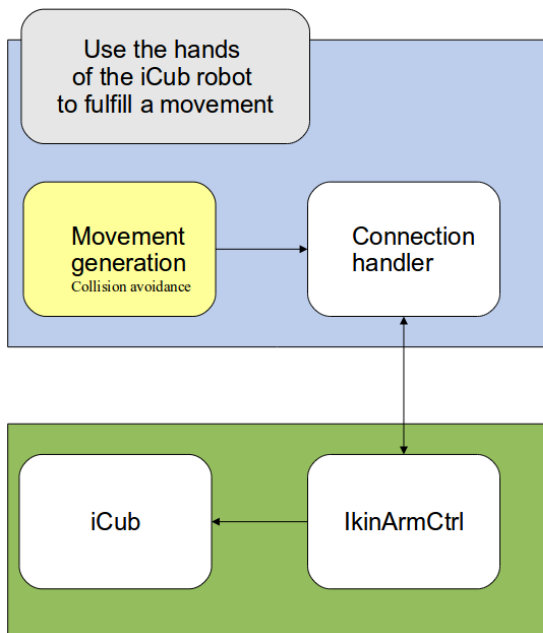
needed
to be
programmed

already
available

Demo



Movement generation



needed
to be
programmed

already
available

How to follow a trajectory?

Trajectory:

- ▶ user defined (circle)

How to follow a trajectory?

Trajectory:

- ▶ user defined (circle)

Controller:

1. right/left arm
2. both arms

How to follow a trajectory?

Trajectory:

- ▶ user defined (circle)

Controller:

1. right/left arm
2. both arms

Smooth trajectory

- ▶ get actual end-effector position
- ▶ check if end-effector is near your current target
- ▶ → send next target point
- ▶ collision avoidance

How to follow a trajectory?

Trajectory:

- ▶ user defined (circle)
- ▶ generated by an external module (optional)

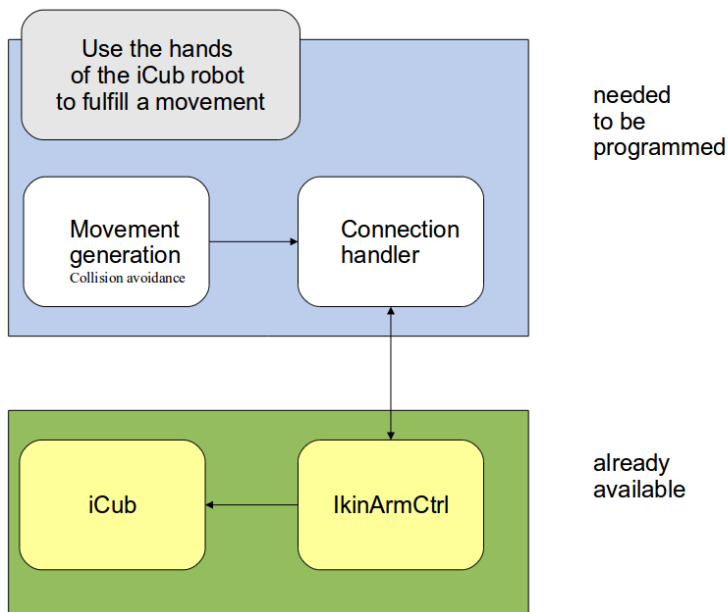
Controller:

1. right/left arm
2. both arms

Smooth trajectory

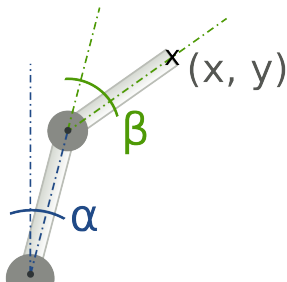
- ▶ get actual end-effector position
- ▶ check if end-effector is near your current target
- ▶ → send next target point
- ▶ **collision avoidance**

Movement generation

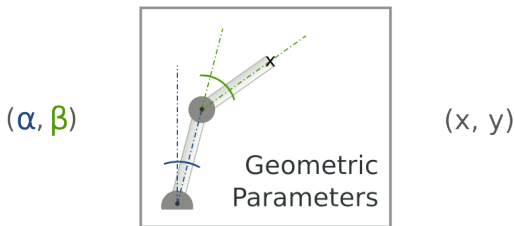


Controlling iCub – Task Space / Joint Space

- ▶ Trajectory in Task Space (Cartesian Coordinates)
- ▶ Control the Robot Joints (Joint Space)

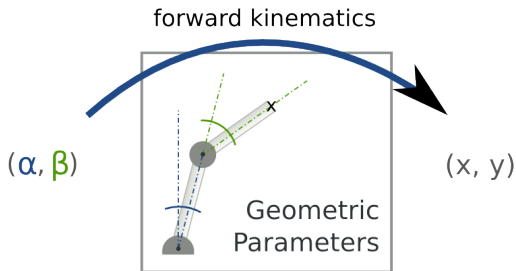


Controlling iCub – iKin



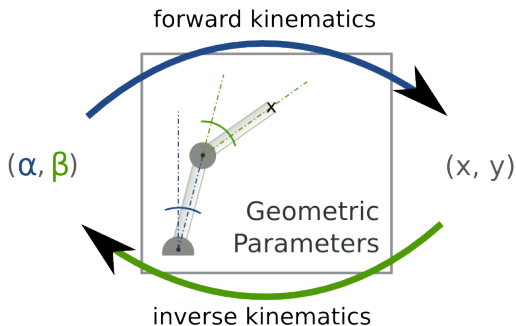
- ▶ Forward Kinematics: *“Where is the endeffector?”*
- ▶ Inverse Kinematics: *“How to move the joints?”*

Controlling iCub – iKin



- ▶ Forward Kinematics: *“Where is the endeffector?”*
- ▶ Inverse Kinematics: *“How to move the joints?”*

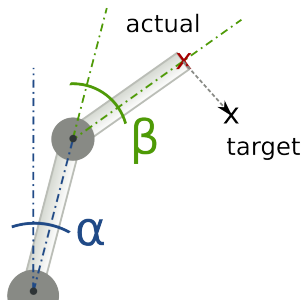
Controlling iCub – iKin



- ▶ Forward Kinematics: *"Where is the endeffector?"*
- ▶ Inverse Kinematics: *"How to move the joints?"*

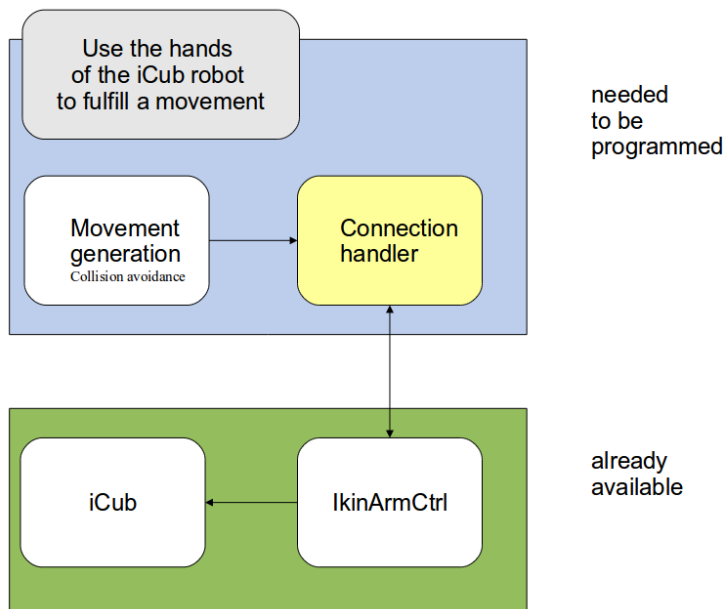
Controlling iCub – iKinArmCtrl

*Prepared classes for iCub limbs, e.g.: **iKinArmCtrl***



- Provides
 - **Solver:** Forward kinematics, Inverse kinematics
 - **Controller:** Comparison of set and actual value, Control

Controlling iCub – Task Space / Joint Space



Connection Handler

iKinArmCtrl ports:

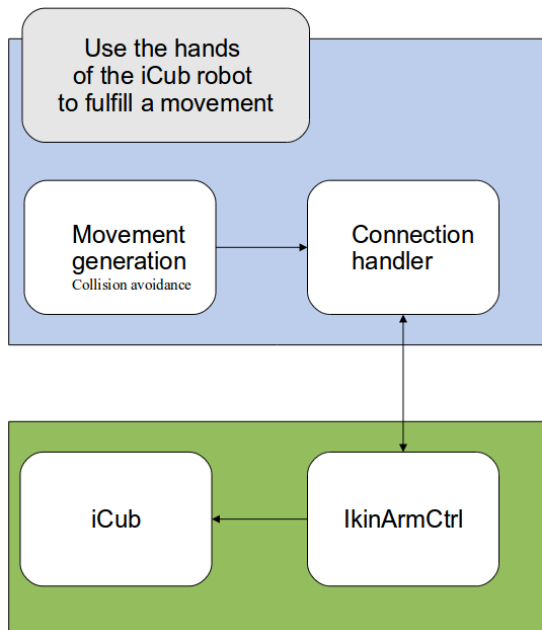
- ▶ /iKinArmCtrl/right or left_arm/command:o
- ▶ /iKinArmCtrl/right or left_arm/q:o
- ▶ /iKinArmCtrl/right or left_arm/qd:o
- ▶ /iKinArmCtrl/right or left_arm/rpc
- ▶ /iKinArmCtrl/right or left_arm/rpc:o
- ▶ /iKinArmCtrl/right or left_arm/simpleState:o
- ▶ /iKinArmCtrl/right or left_arm/state:i
- ▶ /iKinArmCtrl/right or left_arm/torso/command:o
- ▶ /iKinArmCtrl/right or left_arm/torso/rpc:o
- ▶ /iKinArmCtrl/right or left_arm/torso/state:i
- ▶ /iKinArmCtrl/right or left_arm/v:o
- ▶ /iKinArmCtrl/right or left_arm/x:o
- ▶ /iKinArmCtrl/right or left_arm/xd:i

/iKinArmCtrl/right or left_arm/xd:i

► Bottle description:

1. The z axis of the root reference frame is parallel to gravity but pointing upwards.
2. The x axis of the root reference frame points behind the robot.
3. The y axis of the root reference frame points laterally and is chosen according to the right hand rule.
4. - 6. the next 3 values are describing the norm vector of the hand
7. the last value is the rotation of the hand

The Job



needed
to be
programmed

already
available